

## Articles

### Don't waste your Time with good Ideas

If you look at [news.ycombinator](#) at least once a day, you will occasionally see the life guide articles. Many are good and help you to look at something from a different angle. When I was working on my internet video problem, it also developed into a life guide because I made mistakes (due to my habit as a programmer). It's not really bad now and as you get older you automatically stop trying out new things. If you don't quite understand what I mean, please read all of [Terry Pratchett's](#) books. They will help you with my thought processes.

People with a good self-confidence (I'm just touching my own nose) tend to make life unnecessarily difficult for themselves. People work for hours on hundreds of good ideas and at some point they lose sight of their goal. It gets more and more complex and at some point you have written your own framework in Python. Why? Fuck! Because I can do it easily! That is really stupid. You finish diagrams, a flowchart, you sort, brainstorm, discuss and wander deeper and deeper into the forest of your own intelligence (or your own special knowledge) until you come to your grandmother's house, where the reality wolf is waiting for you. You are incredibly creative and have a thousand brilliant ideas? I assure you 999 of them suck.

One of these wonderfully crappy ideas is to be able to watch videos in a browser. I don't know who started it, but in the last few years it's just grown out of place. Browsers can play (or display) more and more formats and are getting bigger, slower and slower. Fortunately, there aren't quite as many developers who use Flash as their video format anymore, and even the whole Microsoft formats are only rarely in use. [HTML5](#) could, besides Javascript, assert itself for this idea. Both are shitty. Because everything runs in the browser and has to be loaded when building the website. That takes time and I personally have no patience for it. Even though I'm working on the Alligator project, I hadn't taken care of it yet.

Now I had watched this week an article by Aral Balkan in which a video was embedded. So far I had read the articles with pleasure and also wanted to look at the video material. Unfortunately my very simple browser couldn't do that yet, so I started to develop really good ideas. See the hint. Well, one of them was to teach my software what HTML5 or Javascript is. Then that wouldn't be a problem anymore. Unfortunately there are still incredibly many video file formats. I was bothered by the fact that all providers, small or large, have their own webplayer design. Ergo. That's all mixed up. I like the creative web, but with certain things or applications I would like more standards. I researched, read articles, considered and wrote down key points and had long forgotten what I actually wanted. At some point I asked myself the simple question whether all this really had to be in the browser. No.

Actually, I only need one piece of software that fulfills all my wishes. Why should I do all the work for myself and reinvent the wheel when I don't have the time anyway? Then rather (first) use a ready-made solution and implement it into your own project. I had decided to use the [MPlayer](#) first, because I had already used it before. But I had to find out that it was replaced by the [mvp](#). Both are [open source](#) and can be installed with a simple `sudo apt-get $software_name` on a [Linux Mint](#) system. Since my browser was written in [Python](#), I only had to tell it what to do when opening a video link. That shouldn't be so difficult.

If you haven't downloaded the [Alligator Browser](#) yet or don't want to, a short explanation how I built the project. So far the browser consists of only two files, the `config.py` and the `alligator.py`. In the first file I will include everything that should be managed by the user himself. I want to keep it as simple as possible, so that nobody gets a headache or has to study computer science before. I hate complicated software that can't be explained by itself. In the second is the actual programming code. So we tell the config file to use the mpv player.

```
videoPlayer = 'mpv '
```

Even if programmers like to present themselves as magicians, there is nothing special about it. It's like riding a bike, once you understand it, it's pretty easy. Now we have to provide all the rest in the second file (alligator.py). Also here there are not many lines of source code. Altogether we say: \*If a user clicks on a link with the right mouse button open a small menu. A menu item should read Open with Videoplayer. When this is pressed, the program reads the link and combines it into a single command with the string (which starts our mediaplayer) in the `config.py` file. The command is executed from the [terminal](#). In Linux you can start software from the [terminal](#).

```
def contextMenuEvent(self, event):
    [...]
    openVideo = (config.videoPlayer)

    if url.isValid():
```

```
[...]
openVideo = menu.addAction("Open with Videoplayer")

else:
    [...]
    openVideo = ""
    [...]

elif action == openVideo:
    os.system(config.videoPlayer + url.toString())
```

I only had to write five lines of code and saved myself a huge amount of work. mvp can open almost everything and even if I want to access the download link via an HTML button this is no problem now. Of course people have to provide me with a link to their video file. But since my browser is developed according to the rules of digital evolution, I simply exclude all other possibilities. It is easy to include a link in your source code than to have to deal with Javascript first. For a preview, a screenshot of the current video can be taken. Thus your side is loaded also faster and one leaves to the user which player he wants to use most. So everyone has profited from it and can go to bed happier in the evening. To test everything I visited the website of [Tears of Steel](#) and called up a download link there. It worked all without problems and the loading times were really minimal. Instead of concentrating on hundreds of genius ideas, just orientate yourself to the simplest solution.